

ממ"ן 14 – שלד פתרון

הקורס : 20471 (ארגון המחשב) 2023א

שאלה 1 (35%)

שלב חמישי (WB) :

הפקודה היא : $\$, \$?, \$12$??? (פקודה מפורמט R)

נימוק :

• לפי ערכי הבקרה וטבלה 4.49, נסיק כי הפקודה הנמצאת בשלב זה **כותבת** למקבץ

האוגרים תוצאה מה**ALU** ולא מהזיכרון, כלומר מדובר בפקודה מפורמט R.

• הפקודה כותבת את התוצאה לאוגר מספר 12.

מיקומה בזיכרון : $0x1853FFF0$

נימוק :

נתונה לנו כתובת הפקודה הנמצאת בשלב 1 (הערך הזורם על קו $PC+4$ בשלב 2). פקודה

זו נמצאת 4 פקודות לפני פקודה זו ובהתאם הכתובת שלה תהיה 16 בתים לפני כתובת

הפקודה הנמצאת בשלב 1, זאת משום שלא התבצע branch.

(אסביר בהרחבה בשלבים הרלוונטיים).

קידודה ב-8 ספרות הקסא: לא ניתן לדעת, משום שלא נתונים לנו אוגרי המקור rs , וכן הערך הנתון ב $funct$.

אילו ערכים מכילה/מחשבת/כותבת לאן ?

הפקודה כותבת ערך מסוים לאוגר מספר 12, שנקרא באותו מחזור שעון (חציית מקבץ)

האוגרים) ביציאה מספר 2 של מקבץ האוגרים, זאת משום שאוגר rt של הפקודה הנמצאת

בשלב 2 הוא גם אוגר מספר 12 (אסביר בהרחבה בשלב הרלוונטי). לכן, מתבצע:

$$\$12 \leftarrow 0xD$$

שלב רביעי (MEM) :

הפקודה היא : $\$, \$?, \$23$??? (פקודה מפורמט R)

נימוק :

• לפי ערכי בקרת WB, גם כאן מדובר בפקודה מפורמט R.

• שדה rd בפקודה הוא האוגר שמספרו $0x17$, כלומר אוגר מספר 23.

• כניסת הכתיבה לזיכרון (שלא ממומשת) - ערכה תוכן אוגר 3. משמעות הדבר היא ששדה rt בפקודה הוא אוגר 3.

מיקומה בזיכרון : $0x1853FFF4$

נימוק :

הפקודה הקודמת אינה פקודת קפיצה, ולכן הפקודה בשלב זה נמצאת ברצף אחריה בזיכרון וכתובתה גדולה ב-4.

קידודה ב-8 ספרות הקסא: לא ניתן לדעת, משום שחסרים לנו שדות funct והrs של הפקודה.

אילו ערכים מכילה/מחשבת/כותבת לאן ?

ניתן שעל הקו המכיל את התוצאה שיצאה מהALU בשלב הקודם (ותיכתב לאוגר היעד

בשלב הבא) זורם הערך 0x5690. מכאן שבמחזור השעון הבא יתבצע:

$$\$23 \leftarrow 0x5690$$

שלב שלישי (EXE) :

הפקודה היא : `beq $4, $12, ????`

נימוק :

- הערך הנכנס לALU Control הוא הערך 01bin. משמעות הדבר היא שמדובר בפקודת branch.
- בכניסה הראשונה לALU (הערך המתאים לאוגר rs שנקרא בשלב הקודם) נמצא ערך אוגר מספר 4. משמעות הדבר היא שrs=4.
- הערך הזורם על הקו ID/EX. Register Rt הוא 0xC. משמעות הדבר היא שאוגר rtz בפקודה הוא אוגר מספר 12.
- לא ידוע לנו על פי נתוני השאלה ערך ה immediate.

מיקומה בזיכרון : 0x1853FFF8

נימוק :

הפקודה מופיעה ברציפות בזיכרון אחרי הפקודה שלפניה, שלא ביצעה קפיצה.

קידודה ב-8 ספרות הקסא: לא ניתן לדעת באופן מלא, אבל כן באופן חלקי ללא שדה ה immediate שבפקודה. מקבלים, `opcode=0x4=000100`, `rs=0x4=00100`, `rt=0xC=01100`, ולכן נקבל:

0x108C????

אילו ערכים מכילה/מחשבת/כותבת לאן ?

- הפקודה מחשבת בשלב זה את ההפרש בין הערך באוגר 4, שלפי נתוני השאלה הינו

0xFFFFF8C, לערך באוגר 12.

- לא נוכל לדעת מה הערך שנקרא בפועל ממקבץ האוגרים עבור אוגר 12 במחזור השעון הקודם, אך בהנחה ונפתרים כל סיכונים הנתונים, ערכו הנכון של האוגר הוא ערכו המעודכן, לאחר ביצוע הפקודה שכעת נמצאת בשלב 5 (סיכון נתונים זה ייפתר ע"י קידום הערך). לכן, הערך שיושווה בפועל בALU הוא הערך של האוגר כפי שנקרא בשלב 2, 0xD.

- לא מתקיים שוויון ולכן לא תתבצע קפיצה והפקודה הבאה ברצף תהיה בכתובת רציפה לכתובת זו.

שלב שני (ID):

הפקודה היא: $sub \$24, \$4, \$12$.

נימוק:

- על קו opcodes הזורם בשלב זה זורם הערך 0, כלומר זוהי פקודה מפורמט R.
- אוגר הrs הוא אוגר מספר 4 לפי ערך הקו שזורם בכניסה למקבץ האוגרים.
- הערך שייכתב בסיום מחזור השעון לאוגר $ID/EX. RegisterRt$ הוא 0xC, וזהו מספרו של אוגר rt.
- שאר 16 הסיביות הימניות בפקודה 0 – 15 נתונות לנו בקו הזורם לעבר יחידת ה sign extension וערכם 0xC622, ובבינארי 1100 0110 0010 0010. מכאן נקבל $rd = 11000bin = 24$, $shamt = 11000bin = 24$, $funct = 100010bin = 0x22$ ומדובר בפקודת subtract.

מיקומה בזיכרון: $0x1853FFFC$.

נימוק:

היות ואינה פקודה קופצת, כתובתה היא 4 בתים לפני הכתובת הנתונה ע"י ערך הקו $PC+4$ בשלב זה, שהוא $0x18540000$. אם נחסר 4, נקבל את תשובתנו.

קידודה ב-8 ספרות הקסא:

$opcode = 000000$, $rs = 4 = 00100$, $rt = 12 = 01100$ ו16 הביטים הנוספים נתונים לנו. נקבל:

$0x008CC622$

אילו ערכים מכילה/מחשבת/כותבת לאן?

לפי נתוני השאלה לגבי ערכו של rs והערך הנקרא rt (שהינו הערך המעודכן מאחר ו-2 הפקודות לפני פקודה זו לא משנות את ערכו), הפעולה תבצע את החישוב

$$\$4 - \$12 = 0xFFFFF - 0xD = 0xFFFFFEF$$

ויתבצע $0xFFFFFEF \leftarrow \$24$ בסוף 3 פעימות שעון לאחר הפעימה הנוכחית.

שלב ראשון (IF):

הפקודה היא: $addi \$18, \$17, -11$

נימוק:

קידוד הפקודה להקסא נתון לנו במפורש:

0010 0010 0011 0010 1111 1111 1111 0101

ובסידור אחר: 001000 10001 10010 1111 1111 1111 0101

- שדה ה-6, opcode, הסיביות השמאליות של הפקודה, יהיה

$opcode = 001000 = 0x8$. לכן, זוהי פקודת addi.

- שדה ה-rs נתון לנו בפקודה והוא 17. שדה ה-rz נתון לנו בפקודה והוא 18.

- שדה ה-immediate נתון לנו בפקודה והוא $0xFFF5$, כלומר 11 –.

מיקומה בזיכרון : $0x18540000$

נימוק :

הפקודה נמצאת ברציפות בזיכרון לאחר הפקודה שלפניה שלא מבצעת קפיצה.

קידודה ב-8 ספרות הקסא: נתון לנו במפורש, והוא $0x2232FFF5$

אילו ערכים מכילה/מחשבת/כותבת לאן ?

מאחר והפקודה הנמצאת כעת בשלב ה-MEM כותבת את הערך $0x5690$ לאוגר שמספרו

17, הפקודה תוסיף 11 – לערך זה ונקבל:

$\$18 \leftarrow 0x5685$

שאלה 2 (35%)

1. מלאו את הטבלה בהתאם להנחיות בסעיף :
במידה ולא ניתן לדעת ערך בקו מסוים יש לסמן X .

הקו	שם הקו נסו לתת שם המסביר את משמעות הקו	ערכו בבסיס הקסא
A1	הערך המספרי המייצג את הפקודה המקודדת, כפי ששמור בזיכרון.	הפקודה (\$6) \$6, 100 lw כפי שמקודדת בזיכרון - 0x8cc60064
A2	הערך הנוכחי של PC, כלומר כתובת הפקודה הנוכחית בזיכרון	0x00013b5c
A3	תוצאת הפעולה שביצע ALU.	בפקודת load הALU מבצע פעולת חיבור בין הערך המיידי (100) לבין תוכן אוגר המקור (\$6) על מנת לחשב את הכתובת. במקרה זה, תוכן האוגר יהיה הערך שנכתב אליו במחזור השעון הקודם, שהוא \$8+\$7 (ערך כל אחד הוא מספרו כפול 0x1500). בסך הכל נקבל - $0x1500 \cdot 7 + 0x1500 \cdot 8 + 0x64 = 0x13B64$
A4	הערך שאמור להיכתב לאוגר היעד בפקודה שאכן כותבת למקבץ האוגרים	במקרה שלנו, מדובר בערך שיצא מהזיכרון בכתובת אותה חישבנו לעיל. לא נתון לנו מה ערכו ולכן הערך הזורם עליו לא ידוע - X
A5	ה"funct" של הפקודה, 6 הביטים הכי ימניים.	במקרה שלנו מדובר ב6 הביטים הימניים של הערך המיידי, ובהקסא 0x24.

2. סיכוני נתונים:

סיכון נתונים מטיפוס: 1a
בין פקודה 1 בקוד לפקודה 2 בקוד על אוגר מספר 6.
הסיכון נפתר ע"י קידום הערך המעודכן של האוגר כאשר פקודה מספר 2 נמצאת בשלב EXE ופקודה מספר 1 נמצאת בשלב MEM.

סיכון נתונים מטיפוס: load-use+2b
בין פקודה 2 בקוד לפקודה 3 בקוד על אוגר מספר 6
הסיכון נפתר ע"י איתור הסיכון על ידי הHDU, מניעת הכתיבה לאוגר IF/ID ולאוגר הPC, והכנסת "בועה" במקום פקודה מספר 3 כאשר היא נמצאת בשלב ID (ע"י איפוס אותו הבקרה).
לאחר מכן נוצר סיכון נתונים מסוג 2b והוא נפתר ע"י קידום הערך המעודכן של האוגר כאשר פקודה מספר 3 נמצאת בשלב EXE ופקודה מספר 2 נמצאת בשלב WB.

נציין שהערך הנקרא ממקבץ האוגרים כאשר פקודה 3 נמצאת בשלב 2 (לאחר הבועה) הוא הערך של אוגר 6 כפי שנכתב בסיום פקודה 1, זאת עקב חציית מקבץ האוגרים. אילו היו 2 פקודות אחרות בין פקודה 1 ופקודה 3 במקום `lw` ו`sc`, שאינו היו משפיעות על ערכו של אוגר 6, אז היינו מקבלים סיכון נתונים שהיה נפתר ע"י חציית מקבץ האוגרים.

3. בתרשים 6 סימונים C1-C6 על קווים מסוימים על המעבד. יש למלא בטבלה את הערכים על קווים אלו. יש להציג **בבסיס הקסא** ערך לא ידוע יש לסמן ב X.

הקו	שם הקו נסו לתת שם המסביר את משמעות הקו	ערכו בבסיס הקסא
C1	הערך השמור באוגר שמספרו נקבע ע"י כניסת <code>rs</code> , כפי שהיה לפני הפקודה הנמצאת כעת בשלב MEM אילו היינו מריצים במעבד חד-מחזורי. (חציית מקבץ האוגרים תגרום לכך שהערכים שאמורים להיכתב בשלב ה WB יהיו מעודכנים בעת קריאה מהאוגרים).	ערך אוגר 6 כפי שהיה לפני ביצוע פקודת ה <code>load</code> , והוא, בדומה לסעיף א, $0x1500 \cdot 7 + 0x1500 \cdot 8 = 0x13B00$
C2	הערך השמור באוגר שמספרו נקבע ע"י כניסת <code>rt</code> , כפי שהיה לפני הפקודה הנמצאת כעת בשלב MEM אילו היינו מריצים במעבד חד-מחזורי. (חציית מקבץ האוגרים תגרום לכך שהערכים שאמורים להיכתב בשלב ה WB יהיו מעודכנים בעת קריאה מהאוגרים).	ערך אוגר 2 כפי שהיה לפני ביצוע פקודת ה <code>load</code> , והוא לא השתנה מתחילת התכנית: <code>0x2A00</code> .
C3	הערך השמור באוגר הנקבע ע"י כניסת <code>rs</code> , כפי שהיה לפני הפקודה הנמצאת כעת בשלב WB אילו היינו מריצים במעבד חד-מחזורי (במחזור השעון הקודם, כאשר נקראו הערכים, ההסבר היה זהה להסבר שסיפקנו לעיל).	ערכו המקורי של אוגר 6, והוא <code>0x7E00</code> .
C4	הערך השמור באוגר הנקבע ע"י כניסת <code>rs</code> , כפי שהיה לפני הפקודה הנמצאת כעת בשלב WB אילו היינו מריצים במעבד חד-מחזורי (במחזור השעון הקודם, כאשר נקראו הערכים, ההסבר היה זהה להסבר שסיפקנו לעיל).	ערכו המקורי של אוגר 2, והוא <code>0x2A00</code> .
C5	ערכו העדכני של אוגר 2, כפי שהיה לפני הפקודה הנמצאת כעת בשלב EXE אילו היינו מריצים במעבד חד-מחזורי (סיכונים נפתרו ע"י forwarding)	הפקודות שלפני הפקודה הנוכחית לא שינו את ערך אוגר מספר 2, ולכן שוב מדובר בערכו המקורי <code>0x2A00</code> .
C6	ערכי הבקרה של הפקודה הנמצאת בשלב EXE	מאחר ומדובר ב "בועה", איפסנו את קווי הבקרה שלה וכולם אפסים. נקבל <code>0x0</code> .

שאלה 3 (30%)

1. תקלה על קו EX.Flush הגורמת לכך שהמרבב המחובר אליו יעביר תמיד את הערך 0.
מספר הסיביות על הקו: 1
התקלה תגרום לכך שבאופן קבוע הוראות הנמצאות בשלב 3 ישנו את ערכי בקרת הזיכרון שלהם ל-0, ולא תתאפשר קריאה או כתיבה לזיכרון.
דוגמא למצב שהתקלה תשפיע:
בפקודת load למשל, אות ה MemRead יהיה כבוי ולכן יצא מהזיכרון ערך לא צפוי, שאינו הערך הנשמר בכתובת אותה רצינו לקרוא. התוצאה תהיה כתיבה של ערך לא נכון אל מקבץ האוגרים.
2. תקלה בקו IF/ID.write (היוצא מיחידת ה HDU) הגורמת לכך שתמיד ייכתב ערך לאוגר הצנרת IF/ID (כלומר תמיד במצב Enable).
מספר הסיביות על הקו: 1
התקלה תגרום לכך שלא נוכל לחסום כתיבת ערכים אל אוגר הצנרת, כלומר ימשיכו ב"מורד" הצנרת פקודות שלא היינו מעוניינים שימשיכו להתבצע, הן באופן זמני (למשל בסיכון load-use) והן באופן קבוע (למשל בקפיצה).
דוגמא למצב שהתקלה תשפיע:
כאשר נרצה להשתמש בפקודה מסוימת, נניח add, שערך אחד מאוגרי המקור שלה אמור להיקרא מהזיכרון בפקודת load שלפניה. ניתקל בסיכון load-use. יחידת ה HDU תמנע מערך ה PC להתקדם אך לא תמנע מהערכים שנקראו בשלב ה IF להיכתב אל אוגרי הצנרת. כתוצאה מכך, פקודת ה add השמורה באוגר הצנרת תידרס ולעולם לא תבוצע. במקומה, תהיה בשלב ה ID הפקודה שאחריה וגם בשלב ה IF, דבר העלול לגרום לביצועה של אותה הפקודה פעמיים!
3. תקלה ביציאה מהמרבב של ה Regdst הגורמת לכך שתמיד הערך היוצא ממרבב זה ערכו יהיה 0.
- מספר הסיביות על הקו: 5** (מספר הסיביות שנדרשות על מנת לברור בין 32 אוגרים).
התקלה תגרום לכך שכל הוראה הכותבת למקבץ האוגרים תנסה לבצע כתיבה לאוגר מספר 0, דבר שאינו אפשרי מבחינת החומרה. במילים אחרות, לא תתאפשר כתיבה למקבץ האוגרים.
דוגמא למצב שהתקלה תשפיע:
שוב בפקודת load, נרצה שהערך שנקרא מהזיכרון ייכתב לאוגר אותו ציינו בפקודה. במקום זאת, לא יתבצע דבר, והערך שיימצא באוגר היה הערך שהיה בו לפני הפקודה, דבר שיפגע בנכונות התכנית!